A Fast Las Vegas Algorithm for Computing the Smith Normal Form of a Polynomial Matrix

Arne Storjohann

and

George Labahn

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Submitted by Richard A. Brualdi

ABSTRACT

A Las Vegas probabilistic algorithm is presented that finds the Smith normal form $S \in \mathbf{Q}[x]^{n\times n}$ of a nonsingular input matrix $A \in \mathbf{Z}[x]^{n\times n}$. The algorithm requires an expected number of $O^{\sim}(n^3 d(d + n^2 \log ||A||))$ bit operations (where $||A||$ bounds the magnitude of all integer coefficients appearing in A and d bounds the degrees of entries of A). In practice, the main cost of the computation is obtaining a non-unimodular triangularization of a polynomial matrix of same dimension and with similar size entries as the input matrix. We show how to accomplish this in $O^{\sim}(n^5 d(d + \log ||A||) \log ||A||)$ bit operations using standard integer, polynomial and matrix arithmetic. These complexity results improve significantly on previous algorithms in both a theoretical and practical sense.

1. INTRODUCTION

The Smith normal form is a diagonalization of a matrix over a principal ideal domain. The concept originated with the work of Smith [17] in 1861 for the special case of integer matrices. Applications of the Smith normal form include, for example, solving systems of Diophantine equations over the domain of entries [4], integer programming [9], determining the canonical decomposition of finitely generated abelian groups [8], determining the

© Elsevier Science Publishing Co., Inc., 1993
655 Avenue of the Americas, New York, NY 10010 0024- 3795/93/$6.00

similarity of two matrices and computing additional normal forms such as Frobenius and Jordan normal forms [5, 12]. A close variant of the Smith normal form, the Smith-MacMillan form for rational functions, plays an important role in linear systems theory [10].

The Smith normal form is well known theoretically but can be difficult to compute in practice because of the potential for rapid growth in the size of intermediate expressions. In a computational setting, the domain of matrix entries is typically the integers $\mathbf{Z}$ or the ring $\mathbf{F}[x]$ of univariate polynomials with coefficients from a field. In this paper we consider the problem of computing the Smith normal form of a square nonsingular matrix over $\mathbf{Q}[x]$. Computing normal forms for matrices over $\mathbf{F}[x]$ where $\mathbf{F}$ is a field of characteristic zero (e.g. $\mathbf{F} = \mathbf{Q}$) poses a double challenge. The size in bits of intermediate expressions — polynomials in $\mathbf{Q}[x]$ — will depend not only on the polynomial degrees but also on the lengths of individual rational number coefficients.

Formally, a matrix S is said to be the Smith normal form of a nonsingular $A \in \mathbf{F}[x]^{n\times n}$ if there exist unimodular matrices U and V such that $UAV = S$ with S being diagonal, each diagonal entry monic, and where $s_{i,i}$ divides $s_{i+1,i+1}$ for $1 \leq i \leq n-1$. (An $n \times n$ matrix U is said to be unimodular if U is invertible over the domain of entries; if U is over $\mathbf{F}[x]$, then U is unimodular if and only if $\det(U)$ is a nonzero element of $\mathbf{F}$.) The Smith normal form S always exists and is unique. The unimodular matrices U and V (called pre- and post-multipliers respectively) are not unique. The diagonal entries $s_{i,i}$ of S are called the invariant factors of A. The invariant factors are also given by $s_{i,i} = s_i^*/s_{i-1}^*$ where s_i^* is the i-th determinantal divisor of A, that is, the gcd of the determinants of all $i \times i$ minors of A (with $s_0^* = 1$). One can also triangularize rather than diagonalize a polynomial matrix and obtain a related form — the Hermite normal form. A matrix H is said to be the Hermite normal form of a nonsingular $A \in \mathbf{F}[x]^{n\times n}$ if there exists a unimodular matrix U such that $UA = H$ with H upper triangular, each diagonal entry monic, and where off-diagonal entries in each column have smaller degree than the diagonal entry. Proofs for the existence and uniqueness of the Hermite and Smith normal forms can be found in Newman [14, Chapter II] for matrices over a general principal ideal domain

In our work we make a distinction between the problem of computing the Smith normal form S from that of computing the Smith normal form S along with candidates for the pre- and post-multipliers U and V. When working over the domain $\mathbf{Q}[x]$ the problem of computing the Smith normal form with multipliers is fundamentally more difficult (computationally) than only computing the Smith normal form. One can think of the analogy of a related problem, that of solving a Diophantine equation: given

polynomials $u(x)$ and $v(x)$ in $\mathbf{Q}[x]$, solve

$$\sigma(x)u(x) + \tau(x)v(x) = g(x)$$

for $\sigma(x), \tau(x)$ and $g(x) = gcd(u(x), v(x))$. There are a number of algorithms that compute $g(x)$ much more efficiently in the cases when $\sigma(x), \tau(x)$ are not required (see, for example, [6, 15] for a detailed discussions of this issue). Note that such a distinction is not true when computing the Hermite normal form. Given a nonsingular input matrix A and its Hermite normal form H, we can compute the matrix U by $U \leftarrow HA^{\text{adj}}$ where A^{adj}, the adjoint of A, can be found using standard methods.

The main result of this paper is a fast sequential algorithm for computing the Smith normal form of a square nonsingular matrix over $\mathbf{Q}[x]$. The algorithm is probabilistic in the Las Vegas sense — an incorrect result will never be returned but with small probability the algorithm may fail and require repetition. The algorithm is significantly faster than existing algorithms.

The majority of algorithms found in the literature for computing Smith normal forms over $\mathbf{Q}[x]$ are based on first computing the Hermite normal form of a matrix and as such solve the more difficult problem of Smith normal form with multipliers (cf. [12, 13, 20]). Specifically, these algorithms can be used to produce candidates for pre- and post-multipliers U and V such that $UAV = S$ is in Smith normal form within the same asymptotic complexity as they require to produce S alone. One reason for producing multipliers is to verify correctness. In particular, Kaltofen, Krishnamoorthy and Saunders have given a Monte Carlo probabilistic algorithm in [11] that computes the Smith normal form but does not produce pre- and post-multipliers. The drawback of the KKS Monte Carlo algorithm is that it may return an incorrect result which cannot be detected easily.

The only other algorithm that we are aware of that solves for the Smith normal form without multipliers is also given by Kaltofen, Krishnamoorthy and Saunders in [11]. Here, the authors give a proof that computing the Smith normal form over $\mathbf{Q}[x]$ is in $\mathcal{P}$, the class of polynomial time algorithms. Their algorithm uses the fact, a consequence of Kannan [13], that computing the Smith normal form over $\text{GF}(p)[x]$ is in the computational class $\mathcal{P}$. Given a nonsingular $A \in \mathbf{Z}[x]^{n \times n}$, the algorithm computes the Smith normal form of $A \bmod p$ for various primes p and uses Chinese remaindering to reconstruct the Smith normal form of A over $\mathbf{Q}[x]$. This approach is impractical because of the large number of image solutions needed to guarantee correctness.

Our method is based on a similar preconditioning method as used by Kaltofen, Krishnamoorthy and Saunders, followed by a fast non-unimodular triangularization combined with a verification step. Our algorithm is straightforward to implement in a computer algebra system and practical in the

sense that the main cost of the computation is obtaining a non-unimodular triangularization of a polynomial matrix of same dimension and with similar size entries as the input matrix

The rest of this paper is organized as follows. In the next section we give the details of non-unimodular matrix triangularization. Since obtaining such a triangularization is the main cost in our algorithm, we show how to accomplish this step in practice using a homomorphic image scheme to avoid computation with large integers and polynomials. Section 3 gives the probabilistic algorithm. Correctness is proved in Section 4 and the complexity of the algorithm is analyzed in Section 5. The last section summarizes our results and provides directions for future research.

2. NON-UNIMODULAR MATRIX TRIANGULARIZATION

Let $\mathbf{R}$ be a principal ideal domain. A key step in the algorithm of the next section is to compute, for a given $A \in \mathbf{R}^{n\times n}$, a lower triangular matrix $F \in \mathbf{R}^{n\times n}$ such that the matrix $T = FA$ is upper triangular with i-th diagonal entry the determinant of the principal i-th minor of A for $1 \leq i \leq n$. The purpose of this section is to give an explicit definition of the matrices F and T and show how they can be computed efficiently over the domain $\mathbf{R} = \mathbf{Z}[x]$. For our complexity analysis we assume standard polynomial and integer arithmetic. Using standard arithmetic, the cost of multiplying two degree d polynomials from $\mathbf{F}[x]$ is bounded by $O(d^2)$ field operations from $\mathbf{F}$; this also bounds the cost of evaluating a degree d polynomial at $d+1$ distinct points and of interpolating a degree d polynomial through $d+1$ points. Similarly, two $\lceil t \rceil$ bit integers can be multiplied in $O(t^2)$ bit operations; this also bounds the cost of computing either direction of the isomorphism implied by the Chinese remainder theorem where t bounds the total length in bits of the moduli.

First recall some basic definitions and facts from linear algebra. For a matrix $A \in \mathbf{R}^{n\times n}$, the minor M_{ij} of entry a_{ij} is defined to be the determinant of the submatrix obtained by deleting the i-th row and j-th column of A. The cofactor C_{ij} is given by $C_{ij} = (-1)^{i+j} M_{ij}$.

FACT 2.1. *Let A be an $n \times n$ matrix over $\mathbf{R}$ with adjoint A^{adj}. The entries in A^{adj} are given by $A^{\mathrm{adj}}_{ij} = C_{ji}$ for $1 \leq i \leq n$ and $1 \leq j \leq n$. The determinant of A can be written according to the j-th column expansion as*

$$\det(A) = a_{1j}C_{1j} + a_{2j}C_{2j} + \cdots + a_{nj}C_{nj}.$$

LEMMA 2.1. *Let $A \in \mathbf{R}^{n\times n}$ be nonsingular and let $F \in \mathbf{R}^{n\times n}$ be the lower triangular matrix with F_{ij} equal to the cofactor of the element in the*

j-th row, i-th column of the i-th principal minor of A for $1 \leq j \leq i \leq n$. Then, the matrix $T = FA$ will be upper triangular with T_{ij} equal to the determinant of the $i \times i$ minor of A formed from rows $[1, 2, \ldots, i]$ and columns $[1, 2, \ldots, i-1, j]$ for $1 \leq i \leq j \leq n$.

Proof. Follows from Fact 2.1 by noting that the entries of FA are the claimed entries for T — which are determinants of minors of A — written according according to their cofactor column expansion. ∎

Given an $A \in \mathbf{R}^{n \times n}$, we write $A_{1 \cdots i, 1 \cdots j}$ to denote the submatrix comprised of the first i rows and first j columns of A, $\mathrm{FF}(A)$ to denote the matrix F of Lemma 2.1, and $\mathrm{row}(A, k)$ to denote the k-th row of A. For $2 \leq k \leq n$, note that the first k entries in $\mathrm{row}(\mathrm{FF}(A), k)$ are precisely those in the last row of the adjoint of the principal k-th minor of A.

We now show how $\mathrm{FF}(A)$ can be computed in $O(n^3)$ ring operations using a simple variation of fraction-free Gaussian elimination. If fraction-free Gaussian elimination is used to reduce an $A \in \mathbf{R}^{n \times n}$ with rank n or $n-1$ to upper echelon form, and row operations are recorded in a companion matrix F, initially set to be the identity matrix, then $\mathrm{row}(F, n) = s\,\mathrm{row}(A^{\mathrm{adj}}, n)$ where s is 1 if the number of row switches performed during the reduction was even and -1 otherwise. (For a thorough discussion of fraction-free Gaussian elimination, see [6] or the original articles by Bareiss [1, 2].) Similarly, if fraction-free Gaussian elimination is used to zero out entries below the diagonal in the first $k-1$ columns of A, and submatrix $A_{1 \cdots k, 1 \cdots k-1}$ has rank $k-1$, then $\mathrm{row}(F, k) = s\,\mathrm{row}(\mathrm{FF}(A), k))$. The key point here is that if the rank of $A_{1 \cdots k, 1 \cdots k-1}$ is equal to $k-1$, then the reduction up to column $k-1$ can be completed with row switches limited to the first k rows. If the rank of $A_{1 \cdots k, 1 \cdots k-1}$ is less then $k-1$, then this will be detected during the reduction of the first $k-1$ columns because a pivot row will have to chosen with index larger than k. In the latter case, all entries in the last row of the adjoint of $A_{1 \cdots k, 1 \cdots k}$ — which are determinants of $(i-1) \times (i-1)$ minors of $A_{1 \cdots k, 1 \cdots k-1}$ — will be zero. Algorithm 2.1 is a simple variation of fraction-free Gaussian elimination which implements this idea.

ALGORITHM 2.1. `Triangularize`
Input: *An $n \times n$ matrix over $\mathbf{R}$.*
Output: *The matrices $F = \mathrm{FF}(A)$ and $T = FA$.*

(1) [Initialize:]
$d \leftarrow 1$;
$s \leftarrow 1$;
$p \leftarrow 0$;
$B \leftarrow \left[\begin{array}{c|c} A & I_n \end{array}\right]$;
$C \leftarrow$ *the $n \times 2n$ zero matrix;*
(2) [Triangularize:]

for $k = 1$ *to* n *do*
  if $p \leq k$ *then* $row(C, k) \leftarrow s\, row(B, k)$*;*
  for $r = k$ *to* n *while* $T_{rk} = 0$ *do od;*
  if $r > n$ *then break;*
  if $r > k$ *then*
    switch rows k *and* r *of* B*;*
    $s \leftarrow -s$*;*
    $p \leftarrow \max(p, r)$*;*
  for $i = k + 1$ *to* n *do*
    $row(B, i) \leftarrow (B_{kk}\, row(B, i) - B_{ik}\, row(B, k))/d$*;*
  $d \leftarrow B_{kk}$*;*
(3) [Output:] $C = \left[\begin{array}{c|c} T & F \end{array}\right]$*;*

Now consider the problem of computing $F = \mathrm{FF}(A)$ for an $A \in \mathbf{F}[x]^{n \times n}$ that has degrees of entries bounded by $d - 1$. Entries of F and T are determinants of minors of A (up to sign) which are polynomials bounded in degree by nd. We need to assume that $\#\mathbf{F} > nd$ so that we can choose a set $\{x_0, x_1, \ldots, x_{nd}\}$ of distinct evaluation points in $\mathbf{F}$. Let $A|_{x=x_i}$ denote the matrix obtained from A by evaluating each polynomial entry at $x = x_i$. The procedure can now be described as follows: (1) find the matrices $A|_{x=i}$ for $i = 0, \ldots, nd$ at a cost of $O(n^2 \cdot n \cdot d^2)$ field operations; (2) find $F|_{x=i}$ and $T|_{x=i}$ for $i = 0, \ldots, nd$ at a cost of $O(nd \cdot n^3)$ field operations; (3) interpolate the $n^2 + n = O(n^2)$ degree nd polynomial entries in matrices F and T at a cost of $O(n^2(nd)^2)$ field operations. This leads to a cost of $O(n^4 d^2)$ field operations for computing $\mathrm{FF}(A)$ over $\mathbf{F}[x]$.

We can extend this homomorphic imaging scheme to compute $\mathrm{FF}(A)$ for $A \in \mathbf{F}[x]^{n \times n}$ when $\mathbf{F} = \mathbf{Q}$. We start with an input matrix $A \in \mathbf{Z}[x]^{n \times n}$ with degrees bounded by $d - 1$. For p a prime, let $A_p = A \bmod p$ be the matrix in $\mathbf{Z}_p[x]^{n \times n}$ obtained from A by replacing each integer coefficient with its image mod p. To compute $F = \mathrm{FF}(A)$ and $T = FA$ over $\mathbf{Z}[x]$, we find F_p and $T_p = F_p A_p$ over $\mathbf{Z}_p[x]$ for sufficiently many primes p to allow recovery of the integer coefficient appearing in F and T via the Chinese remainder algorithm. Let $||A||$ denote the largest integer coefficient appearing in A. Coefficients of entries of F and T will be integers having magnitude less than $\beta = (\sqrt{n}d||A||)^n + 1$; these have length about $\log \beta = \tilde{O}(n(\log d + \log ||A||))$ bits. The following lemma from Giesbrecht shows that we can choose all our primes to be $l = 6 + \log\log\beta$ bits in length.

LEMMA 2.2. [7] *Let* $x \geq 3$ *and* $l = 6 + \log\log x$*. Then there exist at least* $2\lceil \lceil \log_2(2x) \rceil / (l - 1) \rceil$ *primes* p *such that* $2^{l-1} < p < 2^l$*.*

It follows from this lemma that we can choose a list of $s = 2\lceil \lceil (\log 2\beta) \rceil / (l - 1) \rceil = \Theta((\log \beta)/l)$ distinct primes $(p_i)_{1 \leq i \leq s}$ that are bounded in length by l

bits and that satisfy $\prod_{1\leq i\leq s} p_i > \beta$. The non-unimodular triangularization algorithm can be described as follows: (1) Find the images $(A_{p_i})_{1\leq i\leq s}$; (2) For $1 \leq i \leq s$, compute (F_{p_i}, T_{p_i}) at a cost of $O(s \cdot n^4 d^2 \cdot l^2)$ bit operations using the homomorphic imaging scheme given earlier; (3) Apply Chinese remaindering to recover the $O(n^3 d)$ integer coefficients of F and T at a cost of $O(n^3 d \cdot (\log \beta)^2)$ bit operations. Note that the complexity of step (1) will be bounded by that of step (3). Combining these complexity results we obtain the following.

THEOREM 2.1. *Let $A \in \mathbf{Z}[x]^{n\times n}$ with degrees bounded by $d-1$ be given. The matrices $F = \mathrm{FF}(A)$ and $T = FA$ can be found in in $O\tilde{}(n^5 d(d + \log ||A||) \log ||A||)$ bit operations using standard integer and polynomial arithmetic.*

3. AN ALGORITHM FOR SMITH NORMAL FORM OVER $\mathbf{F}[x]$

In this section we give a fast Las Vegas probabilistic algorithm for computing the Smith normal form of a nonsingular input matrix $A \in \mathbf{F}[x]^{n\times n}$ for the case where pre- and post-multipliers are not also required. Since the diagonal entries of S, the Smith normal form of A, are given by $s_{i,i} = s_i^*/s_{i-1}^*$ where s_i^* is the i-th determinantal divisor of A, it is enough to compute the determinantal divisors of A. One possible method to find s_i^* is to compute the determinants of all $i \times i$ minors of A and set s_i^* to be their gcd. Unfortunately, the number of minors increases exponentially with respect to the matrix dimension. The KKS Monte Carlo Smith Form algorithm of [11] overcomes this problem by preconditioning the input matrix using random unimodular pre- and post-multipliers with entries chosen from a subset of the coefficient field. With high probability, each s_i^* can be determined by taking the gcd of the determinants of only two minors. The drawback of the KKS Monte Carlo Smith Form algorithm is that an incorrect result may be returned.

Our algorithm can be described as follows. Following [11] we first precondition A with random pre- and post-multipliers to obtain a new matrix A' that has the same Smith normal form as A. Using the non-unimodular matrix triangularization algorithm of the previous section, we compute a lower triangular matrix F in $\mathbf{F}[x]^{n\times n}$ such that $T = FA'$ is upper triangular with diagonal entry $T_{i,i}$ being the determinant of the i-th principal minor of A'. The algorithm then sets g_i^* to be the gcd of $(T_{n,n})^2$ and the determinant of the i-th principal minor of A'. With high probability, g_i^* will equal s_i^*, the i-th determinantal divisor of A. The remainder of the algorithm performs $O(n^2)$ divisibility checks which all hold if and only if all the g_i^* are indeed the desired determinantal divisors.

To bound the probability of failure by a constant ϵ, where $0 < \epsilon < 1$, we require that the coefficient field $\mathbf{F}$ has at least $\lceil 6n^3 d/\epsilon \rceil$ elements. Since our main motivation is the case when the coefficient field $\mathbf{F}$ has characteristic zero, for example $\mathbf{F} = \mathbf{Q}$, this will pose no restriction. In any case, if the cardinality of $\mathbf{F}$ is too small we can work over an algebraic extension of $\mathbf{F}$ having the required number of elements. Since the Smith normal form is an entirely rational form, computing over an extension field of $\mathbf{F}$ will not change the result.

In what follows, recall that $s^*(A, i)$ denotes the i-th determinantal divisor of A, that is, the gcd of the determinants of all $i \times i$ minors of A. Similarly, $h^*(A, i)$ will denote the gcd of the determinants of all $i \times i$ minors of the first i columns of A.

ALGORITHM 3.1. `SmithForm`
Input: *A nonsingular matrix* $A \in \mathbf{F}[x]^{n \times n}$.
Output: $[s_1^*, s_2^*, \ldots, s_n^*]$, *the determinantal divisors of* A.
Constant: *An upper bound* $0 < \epsilon < 1$ *on the probability of failing.*

(1) [Randomize:]
Let $d - 1$ *bound the degrees of entries of* A *and let* C *be a subset of* $\mathbf{F}$ *with cardinality* $\lceil 6n^3 d/\epsilon \rceil$.
$U_R \leftarrow$ *a unit upper triangular matrix with off diagonal elements chosen at random from* C;
$V_R \leftarrow$ *a unit lower triangular matrix with off diagonal elements chosen at random from* C;
$A' \leftarrow U_R A V_R$;

(2) [Triangularize:]
$F \leftarrow FF(A')$;
$T \leftarrow FA'$;

(3) [Find probable determinantal divisors of A*:]*
$d^* \leftarrow (T_{n,n})^2$;
for $i = 1$ *to* n *do*
  $g_i^* \leftarrow$ *an associate of* $\gcd(d^*, T_{i,i})$;

(4) [Check divisibility properties of g_i^**'s:]*
$g_0^* \leftarrow 1$;
for $i = 1$ *to* $n - 1$ *do*
  if g_i^{*2} *does not divide* $g_{i-1}^* g_{i+1}^*$ *then FAIL;*

(5) [Ensure that $g_i^* = h^*(A', i)$ *for* $1 \le i \le n$*:]*
for $i = 2$ *to* n *do*
  for $j = 1$ *to* $i - 1$ *do*
    if g_{i-1}^* *does not divide* $F_{i,j}$ *then FAIL;*

(6) [Ensure that $g_i^* = s^*(A', i)$ for $1 \le i \le n$:]
for $i = 1$ to $n-1$ do
for $j = i+1$ to n do
if g_i^* does not divides $T_{i,j}$ then FAIL;

(7) [Output:]
$[s_1^*, s_2^*, \ldots, s_n^*]$ with s_i^* the monic associate of g_i^* for $1 \le i \le n$;

4. ALGORITHM CORRECTNESS

In this section we prove that the algorithm `SmithForm` is a correct Las Vegas algorithm for computing a Smith normal form over $\mathbf{F}[x]$. We will prove this using a number of intermediate lemmas that hold for matrices over general principal ideal domains. In what follows, $\mathbf{R}$ will denote a principal ideal domain and we write $a \simeq b$ to mean that a and b are associates over $\mathbf{R}$.

Our approach requires us to determine a matrix triangularization that has the same diagonal entries as the Hermite normal form of an $A \in \mathbf{R}^{n\times n}$. We note that the i-th diagonal entry $h_i = h(A, i)$ of the (unique) Hermite normal form of A is given by $h_i = h_i^*/h_{i-1}^*$ where $h_i^* = h^*(A, i)$ is the gcd of the determinants of all $i \times i$ minors in the first i columns of A (with $h^*(A, 0) = 1$).

LEMMA 4.1. *Let A and U in $\mathbf{R}^{n\times n}$ be nonsingular with $T = UA$ upper triangular. The following statements are equivalent:*

(1) U is unimodular;

(2) $\det(T) \simeq \det(A)$;

(3) $T_{i,i} \simeq h(A, i)$ for $1 \le i \le n$.

Proof. It follows from the identity $\det(T) = \det(U)\det(A)$ that (1) and (2) are equivalent. To see that (3) implies (2) note that $\prod_{1\le i\le n} h(A, i) = |\det A|$. Now assume that (1) holds. The matrix T can be reduced to a matrix H_T in Hermite normal form using unimodular row operations that keep the diagonal entries in the same associate class (see, for example, [3, proof of Corollary 2.3] or [14, proof of Theorem II.2]). In particular, there exists a unimodular matrix U_T such that $U_T T = H_T$. Since the Hermite normal form of A is unique, H_T must be the Hermite normal form of A since $H_T = (U_T U)A$ where $(U_T U)$ is unimodular. This shows that (1) implies (3). ∎

LEMMA 4.2. *Let $T \in \mathbf{R}^{n\times n}$ be nonsingular and upper triangular with i-th diagonal entry t_i. If t_i divides all off-diagonal entries of row i of T for $1 \le i \le n$, then there exists a unimodular matrix $V \in \mathbf{R}^{n\times n}$ such that $TV = D$ where D is the diagonal matrix in $\mathbf{R}^{n\times n}$ with i-th diagonal entry t_i for $1 \le i \le n$.*

Proof. The matrix $D^{-1}T$ will be unit upper triangular over $\mathbf{R}$, so $V = (D^{-1}T)^{-1}$ is unimodular over $\mathbf{R}$ with $TV = TT^{-1}D = D$. ∎

The divisibility properties of the invariant factors and determinantal divisors provides a necessary (but not sufficient) condition for the correctness of an algorithm that returns a list of candidates for the determinantal divisors of an input matrix. This is made precise by the following lemma.

LEMMA 4.3. *For a principal ideal domain $\mathbf{R}$, let $g_0^*, g_1^*, \ldots, g_n^*$ be nonzero elements from $\mathbf{R}$ with $g_0^* = 1$. Then, there exists a matrix in $\mathbf{R}^{n\times n}$ having, for $1 \le i \le n$, the i-th determinantal divisor an associate of g_i^*, if and only if*

$$g_i^{*2} \mid g_{i-1}^* g_{i+1}^*, \qquad 1 \le i \le n-1.$$

Proof. (*If:*) Assume that $g_i^{*2} \mid g_{i-1}^* g_{i+1}^*$ for $1 \le i \le n-1$. First we show that $g_{i-1}^* \mid g_i^*$ for $i = 1, 2, \ldots, n$. For $i = 1$, $g_0^* = 1$ implies $g_0^* \mid g_1^*$. By induction on i, assume $g_{i-1}^* \mid g_i^*$ for $i = 1, \ldots, k$. Then $g_k^{*2} \mid g_{k-1}^* g_{k+1}^* \Rightarrow (g_k^*/g_{k-1}^*)g_k^* \mid g_{k+1}^* \Rightarrow g_k^* \mid g_{k+1}^*$. Next, let g_i be the monic associate of g_i^*/g_{i-1}^* for $1 \le i \le n$. Then, for $1 \le i \le n-1$, $g_i^{*2} \mid g_{i-1}^* g_{i+1}^* \Rightarrow g_i^*/g_{i-1}^* \mid g_{i+1}^*/g_i^* \Rightarrow g_i \mid g_{i+1}$. This shows that the $n \times n$ diagonal matrix S with i-th diagonal entry g_i is in Smith normal form. Furthermore, S has i-th determinantal divisor an associate of g_i^*.

(*Only If:*) See, for example, Newman [14, §16 of Chapter II]. ∎

LEMMA 4.4. *Let A, $U^{(1)}$ and $U^{(2)}$ be matrices in $\mathbf{R}^{n\times n}$ with A nonsingular and let $g_0^*, g_1^*, \ldots, g_n^*$ be entries of $\mathbf{R}$. If*

(1) $T^{(1)} = U^{(1)}A$ and $T^{(2)} = U^{(2)}A$ are upper triangular matrices;

(2) $g_0^* \simeq 1$ and $g_i^* \simeq \gcd(T_{i,i}^{(1)}, T_{i,i}^{(2)})$ for $1 \le i \le n$;

(3) $g_n^* \simeq \det(A)$;

(4) g_i^* divides each entry in row $i+1$ of $U^{(1)}$ and $U^{(2)}$ for $1 \le i \le n-1$,

then $g_i^* \simeq h^*(A, i)$ for $1 \le i \le n$.

Proof. Let A, $U^{(1)}$, $U^{(2)}$ be matrices and $g_0^*, g_1^*, \ldots, g_n^*$ polynomials that satisfy the conditions of the lemma. Condition (4) implies $g_{i-1}^* \mid \gcd((U^{(1)}A)_{i,i}, (U^{(2)}A)_{i,i}) = \gcd(T_{i,i}^{(1)}, T_{i,i}^{(2)}) \simeq g_i^*$ for $1 \le i \le n$ so $g_0^* \mid$

$g_1^* \mid \cdots \mid g_n^*$, where $g_n^* \neq 0$ since A is nonsingular. This implies that $g_i^* \neq 0$ for all $1 \leq i \leq n$. We show by construction that there exists a matrix $U \in \mathbf{R}^{n \times n}$ such that UA has i-th diagonal entry g_i^*/g_{i-1}^*. The desired result will then follows by Lemma 4.1 and the fact the $\det(UA) = \prod_{i=1}^n g_i^*/g_{i-1}^* = g_n^* \simeq \det(A)$.

Condition (2) implies that there exists a solution (a_i, b_i) to the diophantine equation

$$a_i T_{i,i}^{(1)} + b_i T_{i,i}^{(2)} = g_i^*.$$

Let $E^{(1)}$ and $E^{(2)}$ be diagonal matrices in $\mathbf{R}^{n \times n}$ such that for $1 \leq i \leq n$, $(E_{i,i}^{(1)}, E_{i,i}^{(2)})$ is such solution for (a_i, b_i). Let $G \in \mathbf{R}^{n \times n}$ be diagonal with $G_{i,i} = g_{i-1}^*$ for $1 \leq i \leq n$. Now consider the matrix

$$U = E^{(1)} G^{-1} U^{(1)} + E^{(2)} G^{-1} U^{(2)}.$$

Condition (4) implies that U is over $\mathbf{R}$ (i.e. not just over the quotient field of $\mathbf{R}$), so that UA also has all entries from $\mathbf{R}$. Also,

$$\begin{aligned}
UA &= (E^{(1)} G^{-1} U^{(1)} + E^{(2)} G^{-1} U^{(2)}) A \\
&= G^{-1} (E^{(1)} T^{(1)} + E^{(2)} T^{(2)}) A \\
&= G^{-1} \begin{bmatrix}
g_1^* & a_1 T_{1,2}^{(1)} + b_1 T_{1,2}^{(2)} & a_1 T_{1,3}^{(1)} + b_1 T_{1,3}^{(2)} & \cdots & a_1 T_{1,n}^{(1)} + b_1 T_{1,n}^{(2)} \\
0 & g_2^* & a_2 T_{2,3}^{(1)} + b_2 T_{2,3}^{(2)} & \cdots & a_2 T_{2,n}^{(1)} + b_2 T_{2,n}^{(2)} \\
0 & 0 & \ddots & & \vdots \\
\vdots & & & & \\
0 & 0 & \cdots & 0 & g_n^*
\end{bmatrix}
\end{aligned}$$

Thus UA is upper triangular with $(UA)_{i,i} = g_i^*/g_{i-1}^*$. ∎

We are now in a position to prove that algorithm `SmithForm` never produces an incorrect list for the determinantal divisors of the input matrix.

THEOREM 4.1. *Given an input matrix A, algorithm* `SmithForm` *does not return FAIL if and only if the $(g_i^*)_{1 \leq i \leq n}$ found in step (3) satisfy $g_i^* \simeq s^*(A, i)$ for $1 \leq i \leq n$.*

Proof. In what follows, take A', F, T, d^* and $(g_i^*)_{1 \leq i \leq n}$ to be the quantities computed during one pass of algorithm `SmithForm` with input A. The matrix A' is unimodularly equivalent to A and thus has the same

Smith normal form and determinantal divisors as A. Thus, it is sufficient to prove that FAIL is not returned if and only if $g_i^* \simeq s^*(A', i)$ for $1 \leq i \leq n$.

(*If:*) Assume that $g_i^* \simeq s^*(A', i)$ for $1 \leq i \leq n$. By Lemma 4.3, step (4) will not produce a FAIL. By construction we have that the entries of F will satisfy that $F_{i,j}$ is an associate of an $(i-1) \times (i-1)$ minor of A' for $i > 1$ and $1 \leq j \leq i$. Similarly, $T_{i,j}$ is an $i \times i$ minor of A' for $1 \leq i \leq j \leq n$. Since, by assumption, g_i^* is the gcd of all $i \times i$ minors of A', neither steps (5) nor (6) will produce a FAIL.

(*Only if:*) Assume that the algorithm does not return FAIL. Set $U^{(1)} = \det(A')(A')^{\text{adj}}$ and $U^{(2)} = F$. Then, the success of step (5) ensures that the matrices $U^{(1)}, U^{(2)}, A'$ and polynomials $g_1^*, g_2^*, \ldots, g_n^*$ satisfy the conditions of Lemma 4.4. In particular, the matrix $T^{(1)} = U^{(1)}A'$ will be the $n \times n$ diagonal matrix with diagonal entries equal to d^* (which is equal to $\det(A')^2$), and the matrix $T^{(2)} = U^{(2)}A'$ will be the $n \times n$ upper triangular matrix T. Thus there exists a unimodular matrix U in $\mathbf{R}^{n \times n}$ such that

$$\begin{aligned} UA' &= (E^{(1)}G^{-1}U^{(1)} + E^{(2)}G^{-1}U^{(2)})A' \\ &= G^{-1}(E^{(1)}T^{(1)} + E^{(2)}T^{(2)})A' \\ &= G^{-1}\begin{bmatrix} g_1^* & b_1T_{1,2} & b_1T_{1,3} & \cdots & b_1T_{1,n} \\ 0 & g_2^* & b_2T_{2,3} & \cdots & b_2T_{2,n} \\ 0 & 0 & \ddots & & \vdots \\ \vdots & & & & \\ 0 & 0 & \cdots & 0 & g_n^* \end{bmatrix} \end{aligned}$$

where $E^{(1)}, E^{(2)}, G$ and the $(b_i)_{1 \leq i \leq n}$ are as in Lemma 4.4. Note that UA' has i-th diagonal entry g_i^*/g_{i-1}^*. The success of step (6) implies that UA' satisfies the conditions of Lemma 4.2. Therefore there exists a unimodular matrix V such that $UA'V$ is diagonal with i-th diagonal entry g_i^*/g_{i-1}^* for $1 \leq i \leq n$. Finally, the success of step (4) together with Lemma 4.3 gives the desired result. ∎

It remains to derive a bound on the probability that algorithm `SmithForm` returns FAIL. It is worth noting that all the results of this section up until this point have been proven for matrices over a general principal ideal domain $\mathbf{R}$. In particular, a modification of algorithm `SmithForm` that performed all computations over the ring $\mathbf{Z}$ instead of $\mathbf{F}[x]$ would provide a correct algorithm, in the sense of Theorem 4.1, for computing the determinantal divisors of a square nonsingular integer input matrix. However, to properly bound the probability of failure we require some results that are specific to polynomial domains. The technique follows the same approach used in [12], which shows that the probability of failure is equivalent to the probability that a certain quantity is the root of a multivariate polynomial. For this we make use of three lemmas, the first two from [12]. In

what follows, we write $\text{minor}(A, i)$ to denote the i-th principal minor of A.

LEMMA 4.5. [12, Lemma 3.5] *Let $f_1, \ldots, f_t$ be polynomials in $\mathbf{F}[\rho, x]$, $\bar{\rho}$ is a list of new variables, with $\det f_i \leq e$. Then for some $\bar{e} \leq 2e$, there exists an $\bar{e} \times \bar{e}$ determinant Δ in $\mathbf{F}[\bar{\rho}]$, whose entries are coefficients of f_i, such that for any evaluation $\bar{\rho} \to \bar{r}$ a list of corresponding field elements that are not a root of Δ, $\gcd(f_1(\bar{\rho}), \ldots, f_t(\bar{\rho})) = (\gcd(f_1, \ldots, f_t))(\bar{\rho})$.*

LEMMA 4.6. [12, Lemma3.7] *Let A be a matrix in $\mathbf{F}[x]^{n \times m}$ of rank r and with the degrees of the entries bounded by d, and let $i \in \{1, \ldots, r-1\}$. Then there is a polynomial π_i in $m(m-1)/2$ variables such that if*

(1) V_R in $\mathbf{F}[x]^{m \times m}$ is unit lower triangular,

(2) A_s is the submatrix of AR comprised of the first r columns.

then A_s has rank r, and $s^*(A, i) = h^*(A_s, i)$, unless the $m(m-1)/2$ entries below the diagonal in V_R form a root of π_i. The degree of π_i is not more than $2i^2d + i$.

LEMMA 4.7. *Let A be a matrix in $\mathbf{F}[x]^{n \times m}$ of rank m and with the degrees of the entries bounded by d, and let $i \in \{1, \ldots, m-1\}$. Then there is a polynomial γ_i in $n(n-1)/2$ variables such that if*

(1) $U_R \in \mathbf{F}[x]^{n \times n}$ is unit upper triangular,

(2) d^* is a polynomial with degree less than $2md$ and such that $h^*(A, i)$ divides d^*.

then $h^*(A, i) = \gcd(d^*, \det(\text{minor}(U_R A, i))$, unless the $n(n-1)/2$ entries above the diagonal in U_R together with the $2md$ coefficients of d^* form a root of γ_i. The degree of γ_i is bounded by $4mdi$.

Proof. First consider that case where the matrix U_R contains indeterminants as entries, say $(U_R)_{i,j} = \rho_{i,j}$ for $j > i$ where $\bar{\rho} = (\rho_{i,j})_{1 \leq i \leq n, j > i}$ is a list of indeterminants. From [12, Lemma 3.6], we have that the determinant of the minor is given by $\det(\text{minor}(U_R A, i)) = h^*(A, i)p$, where p is an irreducible polynomial in $\mathbf{F}[x, \bar{\rho}] \backslash \mathbf{F}[x]$ or is 1. Since d^* is independent of the indeterminants $\bar{\rho}$, we must have $h^*(A, i) = \gcd(d^*, \det(\text{minor}(U_R A, i)))$ as required. An application of Lemma 4.5 yields the existence of a $4md \times 4md$ determinant Δ, whose entries are coefficients of x of $\det(\text{minor}(U_R A, i))$ and d^* such that for any evaluation $\bar{\rho} \to \bar{r}$, where $\bar{r}$ is a list of corresponding field elements that are not a root of Δ, $\gcd(d^*, \det(\text{minor}(U_R A, i))) = h^*(A, i)$. It remains to establish a degree bound for Δ. Coefficients of x of $U_R A$ are of degree 1 whence coefficients of x of $\det(\text{minor}(U_R A, i))$ will have total

degrees bounded by i. This leads to a bound on the total degree of Δ of $4mdi$. To complete the proof we set $\gamma_i = \Delta$. ∎

Finally, we are in a position to show that our algorithm computes the Smith form correctly with expected probability.

THEOREM 4.2. *Algorithm* `SmithForm` *is correct and fails with probability less than* ϵ. *The expected cost of finding the Smith normal form of a nonsingular input matrix* A *over* $\mathbf{F}[x]$ *is the cost of one pass of algorithm* `SmithForm`.

To show that the probability of failure is less than ϵ we show that $g_i^* \simeq s_i^*$ for $1 \leq i \leq n$ provided the entries of U_R above the diagonal and V_R below the diagonal do not form the root of a certain polynomial π with degree bounded by $6n^3d$. By a result of Schwartz [16], the probability of this happening is bounded by $\deg(\pi)/\#C$, which by our choice of C is less than ϵ.

The matrix AV_R will be such that $h^*(AV_R, i) = s^*(A, i)$ for $1 \leq i \leq n$ unless the entries of V_R below the diagonal form a root of a polynomial $\pi_S = \pi_1\pi_2\cdots\pi_{n-1}$ where each π_i, bounded in degree by $2i^2d + i$, is as in Lemma 4.6. Similarly, $\gcd(\det(\text{minor}(A', i), \det(A)^2) = h^*(AV_R, i)$ for $1 \leq i \leq n$ if the entries of U_R above the diagonal do not form a root of a polynomial $\pi_H = \gamma_1\gamma_2\cdots\gamma_{n-1}$ where each γ_i, bounded in degree by $4ndi$, is as in Lemma 4.7. The polynomial π_H will be bounded in degree by $4n^3d$ and π_S by $2n^3d$. Let $\pi = \pi_S\pi_H$. Then π is bounded in degree by $6n^3d$.

The probability that k iterations will be required to return a non FAIL result is $\epsilon^{k-1}(1-\epsilon)$. The expected number of iterations required to return a non FAIL result is given by $\sum_{1\leq k\leq\infty} k\epsilon^{k-1}(1-\epsilon)$, which is equal to $1/(1-\epsilon)$, a constant. ∎

5. ALGORITHM COMPLEXITY

Notice first that the entries of F and T found in step (2) are associates of determinants of minors of A'. These have degrees bounded by nd, leading to a bound of $2nd$ on degrees of all intermediate polynomials occurring during the algorithm. Using the evaluation/interpolation scheme discussed in Section 2, the matrices F and T can be found in $O(n^4d^2)$ field operations from $\mathbf{F}$. This bounds the cost of the matrix multiplications in step (1), the n gcd computations in step (3), and the remaining n multiplications and $n^2 - 1$ trial divisions in steps (4), (5) and (6), and leads leads to a bound of $O(n^4d^2)$ field operations for one pass of algorithm `SmithForm`.

Now consider the case when $\mathbf{F} = \mathbf{Q}$. We assume without loss of generality, and as done in [11] and [13], that the input matrix A has been preconditioned to have all coefficients of polynomial entries integral. Although

we are implicitly computing over $\mathbf{Q}[x]$, beginning with input $A \in \mathbf{Z}[x]^{n \times n}$ allows all intermediate computations in steps (1) through (6) to be accomplished over the simpler domain $\mathbf{Z}[x]$. In practice, the dominant cost of the algorithm will almost certainly be finding the triangularization T and transition matrix F in step (2). The integer coefficients appearing in A' will be only slightly larger than those of A. In particular, in step (1) we can choose $C = \{0, \ldots, \lceil 6n^3 d/\epsilon \rceil\}$ so that $||A'|| \leq n \cdot \lceil 6n^3 d/\epsilon \rceil \cdot ||A||$. Asymptotically, the length of integer coefficients in $F = \mathrm{FF}(A')$ and $T = FA'$ will be bounded by $O^{\sim}(n(\log d + \log ||A||))$ bits. By employing the homomorphic imaging scheme developed in Section 2 we have the the following result which follows directly from Theorem 2.1.

THEOREM 5.1. *The cost of a one pass of algorithm* `SmithForm` *with input* $A \in \mathbf{Z}[x]^{n \times n}$ *is* $O^{\sim}(n^5 d(d + \log ||A||) \log ||A||)$ *bit operations using standard integer and polynomial arithmetic plus no more than* $O(n^2)$ *trial divisions, multiplications and gcd computations involving polynomials that are factors of entries in the matrices* F *and* T *found in step (2). Entries of* F *and* T *will be polynomials with degrees bounded by* nd *and with integer coefficients bounded in length by* $O^{\sim}(n(\log d + \log ||A||))$ *bits.*

Next, we derive an asymptotic complexity result for one pass of algorithm `SmithForm`. Let $M(t)$ be an upper bound on the number of bit operations required to multiply two $\lceil t \rceil$ bit integers. Using fast integer multiplication (the Schönhage-Strassen algorithm) we can take $M(t) = t \log t \log \log t$. There is a natural duality between the integers and univariate polynomials with integer coefficients. The integer coefficients (represented in binary) of a degree $d-1$ polynomial $f \in \mathbf{Z}[x]$ having coefficients bounded in magnitude by $2^{k-1} - 1$ ($k \in \mathbf{Z}$) can be written as a binary lineup to obtain the dk bit integer $f|_{x=2^k}$. This corresponds to the B-adic expansion of an integer; choosing B a power of 2 allows the conversion to and from polynomial representation to be accomplished in linear time. Thus, we can find F and T in $O(n^3 M(ndk))$ bit operations by applying fraction-free Gaussian elimination to the $n \times n$ integer matrix $A'|_{x=2^k}$ where $k = \lceil 1 + \log_2(\beta + 1) \rceil$. By a result of Schönhage [15], the n gcd computations in step (3) require an expected number of $O^{\sim}(n \cdot nd(nd + n \log ||A||))$ bit operations. The remaining $O(n^2)$ trial divisions in steps (4), (5) and (6) and the $O(n)$ multiplications in step (3) will require at most $O(n^2 M(nd \cdot (nd + n \log nd||A||)))$ bit operations. Overall this yields $O^{\sim}(n^3 d(d + n^2 \log ||A||))$ bit operations for one pass of algorithm `SmithForm` using fast integer arithmetic.

6. CONCLUSIONS

We have given a Las Vegas algorithm to compute the Smith normal form of a nonsingular polynomial matrix $A \in \mathbf{F}[x]^{n \times n}$ where $\mathbf{F}$ is a field. We have taken a new approach that completely avoids the usual technique of diagonalizing the input matrix with a succession of unimodular row and column operations — this has allowed us to derive very good bounds on the size of intermediate expressions occurring during the algorithm for the case $\mathbf{F} = \mathbf{Q}$. The algorithm is both fast and practical with the main computation being the (non-unimodular) triangularization of a polynomial matrix. The algorithm also works well for matrices over more general polynomial domains, for example extensions of the form $\mathbf{Q}(\alpha)[x]$ where the algebraic number α has a monic minimal polynomial from $\mathbf{Z}[x]$.

A key step in our algorithm — first used in [11] — is to obtain a preconditioning A' of the input matrix $A \in \mathbf{F}[x]^{n \times n}$. A drawback of this technique is that preconditioned matrix A' will be dense even if the input matrix A is sparse. A possible solution to this problem is use sparse preconditioning matrices although we have not investigated this approach.

More recently, though, we have discovered a sequential deterministic version of our probabilistic Smith normal form algorithm that, as well as giving a new complexity result, may be useful in the case of sparse input. The deterministic version constructs a correct premultiplication of the input matrix *during* the non-unimodular triangularization phase of the algorithm. This will be presented in a future paper.

The algorithm we have presented for computing Smith normal forms over $\mathbf{F}[x]$ works only for square nonsingular input matrices. In the future, we will give a generalization that works for both singular and/or rectangular input (see [18] for details). In particular, the generalization to nonsquare matrices depends on a result presented in [19] where we give a fast Las Vegas algorithm for reducing the problem of computing the Hermite normal form of a rectangular polynomial matrix to that of computing the Hermite normal form of a nearly square matrix with similar size entries.

The algorithm we have presented takes advantage of the fact that we did not need to compute candidates for pre- and post-multipliers for the Smith normal form. We also plan to present asymptotically fast algorithms for computing Hermite normal forms over various domains. In particular, the probabilistic algorithm given in [12] for computing the Smith normal form (with multipliers) of a polynomial matrix has as its dominant cost computing Hermite normal forms.

In the case of computing the Smith normal form with multipliers, the multiplier matrices are highly non-unique. An remains an open problem whether this computation can be done to produce "nice" multipliers, that

is, multipliers having small coefficients (when possible). For certain applications it is enough to know that one of the multiplier matrices can be nice. For example, in order to determine if two integer matrices A, B are similar over the rationals, one can compute the Smith normal form of the characteristic polynomial matrices $xI - A, xI - B$. If these normal forms are equal then the matrices are similar. In addition, if $U_A(x), V_A(x)$ and $U_B(x), V_B(x)$ are the multiplier matrices for these Smith forms then a similarity transform matrix T can be computed via $T \leftarrow (V_A V_B^{-1})|_{x=B}$ and will satisfy $B = TAT^{-1}$ (see [5, Chapter VI]). For such an application it is enough to require that the column multiplier matrix have coefficients as small as possible.

REFERENCES

1 E. H. Bareiss. Sylvester's identity and multistep integer-preserving Gaussian elimination. *Mathematics of Computation*, 22(103):565–578, 1968.

2 E. H. Bareiss. Computational solution of matrix problems over an integral domain. *Phil. Trans. Roy. Soc. London*, 10:68–104, 1972.

3 P. D. Domich, R. Kannan, and L. E. Trotter, Jr. Hermite normal form computation using modulo determinant arithmetic. *Mathematics of Operations Research*, 12(1):50–59, February 1987.

4 M. A. Frumkin. An application of modular arithmetic to the constuction of algorithms for solving systems of linear equations. *Soviet Math. Dok.*, 17:1165–1168, 1976.

5 F. R. Gantmacher. *Matrix Theory*, volume 1. Chelsea Publishing Company, 1960.

6 Keith O. Geddes, S. R. Czapor, and George Labahn. *Algorithms for Computer Algebra*. Kluwer, Boston, MA, 1992.

7 Mark Giesbrecht. Fast algorithms for rational forms of integer matrices. In *Proceedings of ISSAC'94*, pages 305–311, Oxford, England, 1994.

8 B. Hartley and T. O. Hawkes. *Rings, Modules, and Linear Algebra*. Chapman and Hall, 1970.

9 T. C. Hu. *Integer Programming and Network Flows*. Addison-Wesley, Reading, MA, 1969.

10 Thomas Kailath. *Linear Systems*. Prentice Hall, Englewood Cliffs, N.J., 1980.

11 Erich Kaltofen, M. S. Krishnamoorthy, and B. David Saunders. Fast parallel computation of Hermite and Smith forms of polynomial matrices. *SIAM Journal of Algebraic and Discrete Methods*, 8:683–690, 1987.

12 Erich Kaltofen, M. S. Krishnamoorthy, and B. David Saunders. Parallel algorithms for matrix normal forms. *Linear Algebra and its Applications*, 136:189–208, 1990.

13 R. Kannan. Polynomial-time algorithms for solving systems of linear equations over polynomials. *Theoretical Computer Science*, 39:69–88, 1985.

14 M. Newman. *Integral Matrices.* Academic Press, 1972.

15 A. Schönhage. Probabilistic computation of integer polynomial GCD's. *Journal of Algorithms*, 9:365–371, 1988.

16 J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27:701–717, 1980.

17 H. J. S. Smith. On systems of linear indeterminate equations and congruences. *Phil. Trans. Roy. Soc. London*, 151:293–326, 1861.

18 Arne Storjohann. Computation of Hermite and Smith normal forms of matrices. Master's thesis, Dept. of Computer Science, University of Waterloo, 1994.

19 Arne Storjohann and George Labahn. Preconditioning of rectangular polynomial matrices for efficient Hermite normal form computation. In *Proceedings of ISSAC'95*, pages 119–125, Montreal, Canada, 1995.

20 Gilles Villard. Computation of the Smith normal form of polynomial matrices. In *Proceedings of ISSAC'93*, pages 208–217, Kiev, Ukraine, 1993.